

POS Reconciliation

Capstone 27 · FinTech · Talodabi Academy

1 · Project overview

Agent daily close: cash vs app balance.

Deliverable	Full mobile MVP in Figma (not code)
Platform	Mobile app · Android-first · portrait
Screens	11 screens + 1 unhappy path (12 frames)
Features	3 core features
Bottom navigation	Today · History · Report · Profile
Prototype	4 linked flows

Build order: Auth (screens 1–4) → nav tabs (5–8) → core flow (9–11) → unhappy path → link prototypes.

2 · Problem & primary user

POS agents reconcile cash vs app balance daily. UK/US agents are rare; Nigeria has millions of mobile money agents.

Primary user

"Chinedu, a 26-year-old Moniepoint agent in Onitsha who closes shop late matching transfers to cash."

Geographic scope: National-level

3 · Goals

User goal: See expected balance vs counted cash and flag mismatches fast.

Business goal: Agent networks and fintechs sell reconciliation tools to agent managers.

Success: User logs day and sees mismatch clearly.

4 · Market context

QUESTION	ANSWER
Who pays	Agent networks and fintechs sell recon tools to super-agents.
Why not WhatsApp	Agents use notebook tallies at close. Errors surface only when balance mismatches.
Design focus	Expected vs counted cash in under 2 minutes on phone.

Competitors to check:

- **Agent app transaction list** (no cash reconcile)
- **Excel by super-agent** (manual)
- **Notebook** (default)

5 • Validate before you design

Interview 3–5 people who match your primary user. Update this brief if interviews contradict it. Do not start hi-fi until you complete the confirmation below.

FIELD	YOUR ANSWER
Validated user (name, age, city)	
Problem in one sentence	
One interview quote	
Success metric for your test	
Primary colour and tone	
One competitor you checked	

Interview prompts

- "Walk me through the last time this happened."
- "What did you use instead? WhatsApp, paper, or nothing?"
- "What would make you trust this on a cheap Android phone?"
- "What would stop you from using this every week?"

6 • Features & scope

v1 • three features

1. Import or log daily transaction totals
2. Enter counted cash and float
3. See mismatch report with suggested checks

Out of scope

- Holding customer funds
- Loans
- Multi-branch enterprise

7 · Screens

Design every row in wireframe, high-fidelity UI, and prototype (where the flow applies).

#	LAYER	SCREEN	WHAT TO SHOW
1	Auth	Splash Screen	App logo, name, and tagline for POS Reconciliation.
2	Auth	Onboarding (1 screen)	Explain the app in one sentence. One Get Started button.
3	Auth	Sign Up Screen	Name, phone number, state, and PIN. No email required.
4	Auth	Sign In Screen	Phone number and PIN. Forgot PIN link.
5	App shell	Today (tab)	Balanced or Mismatch status. Start close button and quick totals. Show bottom nav with Today active.
6	App shell	History (tab)	Past days list with balanced/mismatch badge per day. Show bottom nav with History active.
7	App shell	Report (tab)	Daily summary card with export screenshot button. Show bottom nav with Report active.
8	App shell	Profile (tab)	Agent name, terminal ID, and manager contact. Show bottom nav with Profile active.
9	Core flow	Log day · Saved	Transfers in, cash out, fees entered. Show bottom nav on post-sign-in screens.
10	Core flow	Reconcile · Balanced	Counted cash matches expected balance. Show bottom nav on post-sign-in screens.
11	Core flow	Report · Exported	Daily summary ready for manager screenshot. Show bottom nav on post-sign-in screens.

Layers: Auth = 1–4 · App shell = 5–8 · Core flow = 9–11

8 · Prototype flows

Link these in Figma. Examiners should click through without your voice-over.

Flow 1 · App entry

Splash → Onboarding → Sign Up or Sign In → first main tab

Flow 2 · Core task (all three features)

Sign In → Today tab → Log day → Reconcile → Report tab

Core flow screens: Log day · Saved · Reconcile · Balanced · Report · Exported

Flow 3 · Navigation

Link every nav tab so examiners can tap each bottom nav item. Tabs: Today · History · Report · Profile

Flow 4 · Unhappy path

Trigger the unhappy path edge case and link to that screen.

9 · Unhappy path

Design this as its own screen state (frame 12):

Mismatch over ₦10,000. Show checklist: Check pending transfers, verify last 5 receipts.

Link in **Flow 4**.

10 · Design system

RULE	DETAIL
Colour	You choose. Keep it simple and readable.
Type	You choose. Keep body text easy to read on a phone.
Tone	Design for your user, not for a tech audience.

Reuse on every screen:

- Primary button (active and disabled states)
- Input field (empty, filled, and error states)
- Card or list item used across the app
- Navigation bar with correct labels and icons
- Empty state: what the screen looks like with no data

Deliverables

#	DELIVERABLE	WHAT GOOD LOOKS LIKE
1	Wireframes	Every screen in this brief, plus the unhappy path. Layout and labels clear.
2	High-fidelity UI	Same frames polished. You choose colour and type. Bottom nav on every post-sign-in screen.
3	Prototype	Four linked flows (entry, core task, navigation, unhappy path). Examiners can use the app without your voice-over.
4	AI prototype	Figma Make: at least 2 screens. Share link.
5	Case study (1 page)	Problem · User · Who pays · Competitors · Solution · One design decision
6	Presentation (6–8 slides)	Name · Problem · User · Solution · Key screens · Flow demo · One decision

Before you present

- ☐ Every screen designed in wireframe and hi-fi
- ☐ Unhappy path screen designed and linked in the prototype
- ☐ Four prototype flows linked end to end
- ☐ At least one empty state on a list tab
- ☐ Shared components reused: button, input, card or row, bottom nav
- ☐ Case study names who pays and one competitor you checked
- ☐ You can explain one design decision in critique (why, not what)
- ☐ Completed 5 user interviews or 1 usability test
- ☐ Checked at least one competitor listed under Market context

© Talodabi Academy · For enrolled students · Not for redistribution

